

Docker

A software technology
providing containers

Cosa sono i container?



Cosa sono i container?

- * Un container è un environment di esecuzione completo ed isolato

Cosa sono i container?

- * Un container è un environment di esecuzione completo ed isolato
- * Condivide con il sistema host il kernel



Cosa sono i container?

- * Un container è un environment di esecuzione completo ed isolato
- * Condivide con il sistema host il kernel

Su cosa si basano i container?

Cosa sono i container?

- * Un container è un environment di esecuzione completo ed isolato
- * Condivide con il sistema host il kernel

Su cosa si basano i container?

- * Cgroups
- * Namespaces



Cosa sono i container?

- * Un container è un environment di esecuzione completo ed isolato
- * Condivide con il sistema host il kernel

Su cosa si basano i container?

- * Cgroups
- * Namespaces
- * Docker dalla versione 0.9 include la libreria libcontainer



Container VS Virtual Machine



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida
- un controllo più granulare



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida
- un controllo più granulare

I vantaggi della containerizzazione:

- Più container sul proprio PC



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida
- un controllo più granulare

I vantaggi della containerizzazione:

- Più container sul proprio PC
- Versioning semplificato



Container VS Virtual Machine

Rispetto a un'intera macchina virtualizzata, un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida
- un controllo più granulare

I vantaggi della containerizzazione:

- Più container sul proprio PC
- Versioning semplificato
- Testing più facile



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Immagini:

- sono modelli completi per la creazione di uno o più contenitori



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Immagini:

- sono modelli completi per la creazione di uno o più contenitori
- un'immagine è sostanzialmente un file (o un insieme di file) che è lo snapshot di un contenitore



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Immagini:

- sono modelli completi per la creazione di uno o più contenitori
- un'immagine è sostanzialmente un file (o un insieme di file) che è lo snapshot di un contenitore
- ad es., un'immagine potrebbe contenere un OS Ubuntu con un application server WildFly, un'altra immagine (diversa dalla precedente) potrebbe contenere un OS Ubuntu con Wildfly, insieme a un'applicazione web di interesse



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Immagini:

- sono modelli completi per la creazione di uno o più contenitori
- un'immagine è sostanzialmente un file (o un insieme di file) che è lo snapshot di un contenitore
- ad es., un'immagine potrebbe contenere un OS Ubuntu con un application server WildFly, un'altra immagine (diversa dalla precedente) potrebbe contenere un OS Ubuntu con Wildfly, insieme a un'applicazione web di interesse

Un'immagine è un concetto statico, inerte

- un'immagine non viene eseguita direttamente
- un'immagine non ha un proprio stato
- un'immagine è immutabile



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Un **contenitore** è un'istanza eseguibile di un'immagine Docker

- un contenitore è un concetto dinamico, runtime



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

Un **contenitore** è un'istanza eseguibile di un'immagine Docker

- un contenitore è un concetto dinamico, runtime
- un contenitore può essere eseguito su un host



Di cosa si compone Docker?

Docker si compone principalmente di due elementi:

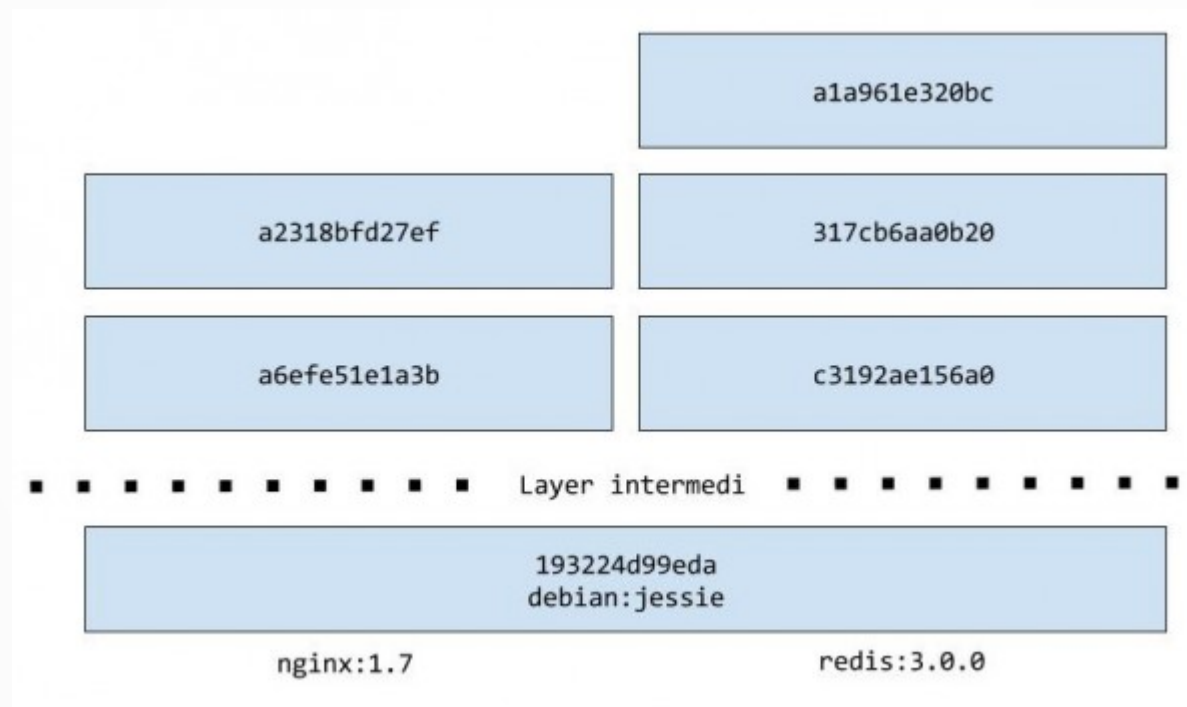
Un **contenitore** è un'istanza eseguibile di un'immagine Docker

- un contenitore è un concetto dinamico, runtime
- un contenitore può essere eseguito su un host
- più precisamente, un contenitore può essere in esecuzione, oppure essere stato arrestato, oppure essere in pausa



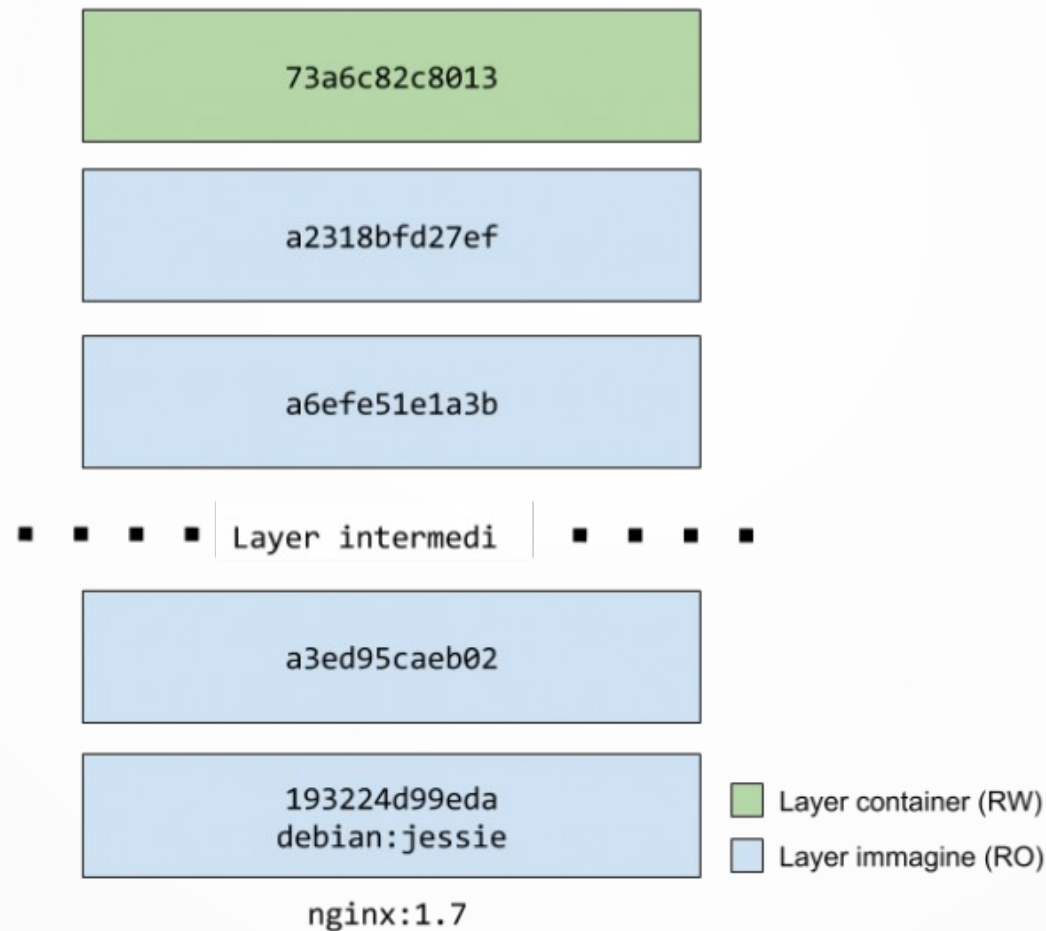
Union filesystem

Le immagini in Docker sono stratificate in layer *immutabili*, ovvero accessibili in sola lettura e non modificabili direttamente



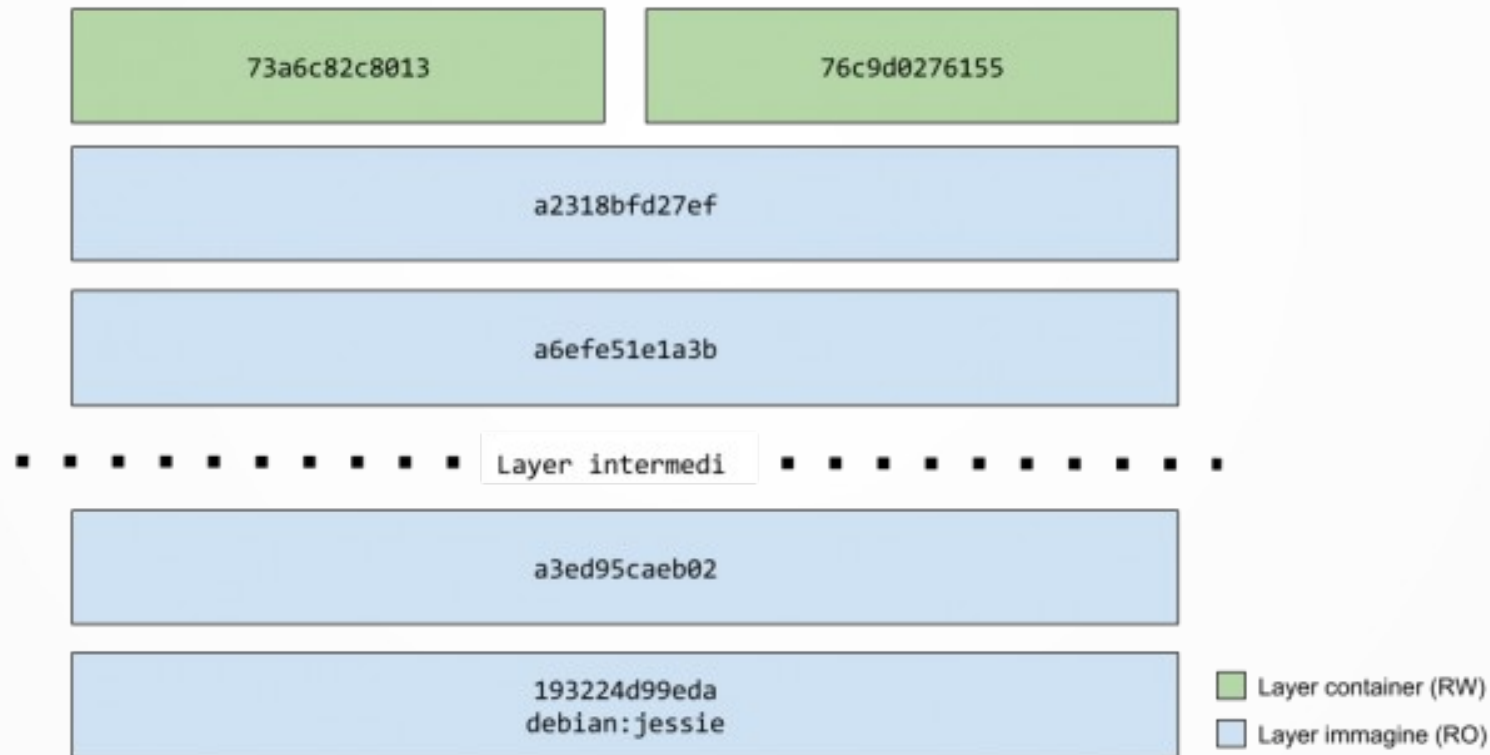
Union filesystem

Tutte le modifiche apportate al container verranno memorizzate all'interno di un nuovo layer, detto anche *layer container*



Union filesystem

Nel momento in cui arriverà la richiesta di avviare una seconda istanza di un container, Docker non farà altro che creare un nuovo layer container.



Esempio esecuzione contenitore Docker

Un primo esempio minimale - basato sull'immagine hello-world disponibile presso il Docker Hub

- `docker run hello-world`

Output:

Hello from Docker!

This message shows that your installation appears to be working correctly.



Costruzioni di immagini

Per la costruzione di immagini personalizzate Docker fa uso di un file chiamato **Dockerfile** con un approccio di tipo infrastructure-as-code

- il Dockerfile contiene tutti i comandi da eseguire per costruire un'immagine personalizzata
- il comando `docker build -t image-name context` consente di costruire automaticamente un'immagine (di nome image-name) a partire da un contesto context



Costruzioni di immagini

Per la costruzione di immagini personalizzate Docker fa uso di un file chiamato **Dockerfile** con un approccio di tipo infrastructure-as-code

- Un Dockerfile è composto da una sequenza di istruzioni

```
# Dockerfile for Apache HTTP Server
FROM ubuntu:14.04
# Install apache2 package
RUN apt-get update && \
    apt-get install -y apache2
# Other instructions
ENV APACHE_LOG_DIR /var/log/apache2
VOLUME /var/www/html
EXPOSE 80
# Launch apache2 server in the foreground
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```



Altri comandi Docker utili

- Per elencare i contenitori in esecuzione
 - `docker ps [-a]`
- per ispezionare le porte usate da un contenitore – utile soprattutto quando si usa l'opzione `-P`
 - `docker port container-name`
 - il risultato è della forma `80/tcp -> 0.0.0.0:8080`
- per ispezionare i log generati in un contenitore
 - `docker logs container-name`

